Layout : generate design layers → fab masks (aka polygon pushing)

DRC design rule checking

Fab generate PDK (process design kit) → loaded in virtuoso

FEOL (front end of the line) : device related (Q)

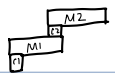
nwell , diff : S, D, plugs , poly : G, interconnect , n(p)imp: (n^+, p^+) , contact between diff & gates

BEOL wires, interconnect

metal 1 via 12 metal 2

↓

v..., m... , lower metal thinner (higher levels for long-dist signal, or V_{DD}/⏚ → R↓, but more space and fringing C)

CMP (chemical mechanical planar polishing) process 

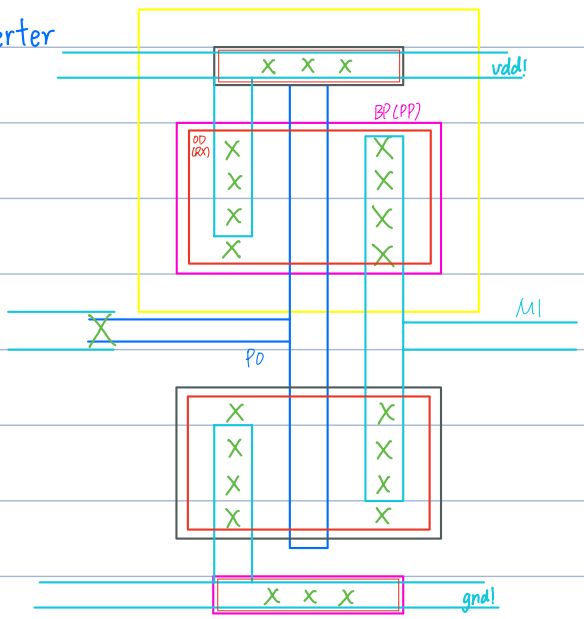
challenge : 1. polishing → e charge (antenna effect) → ~ rules

2. density rules CMP prefers etching SiO_2 → dishing

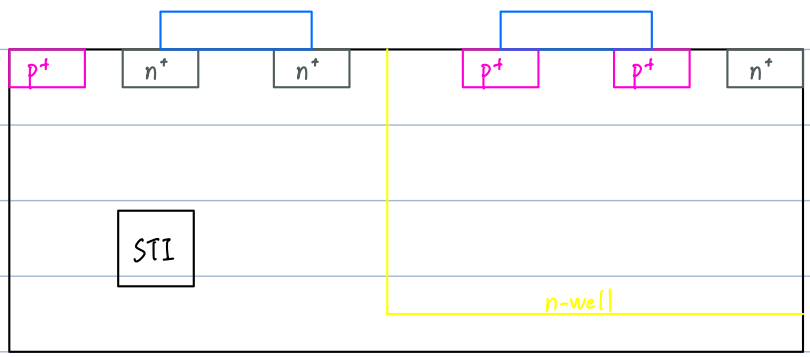
Design rules may not be binary. Soft transition of yield → score

resolution, alignment

E. inverter

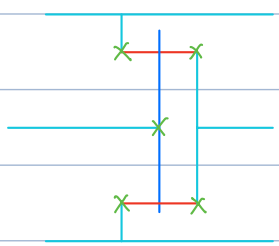


1. nwell (for pfet), preferably shared if multiple pfets
2. V_{DD}!
3. diff (tie nwell w/ V_{DD}!), p/nimp

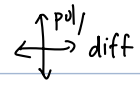


Stick diagrom : layout planning before polygon

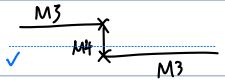
same topology w/ layout



Layout image



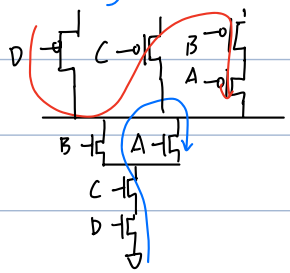
↔ M1 ↕ M2 ↔ M3, every layer ⊥ to the layer below



✓, prevents congested routing

Diffusion sharing → compact layout, reduce C_d, C_{par}

E.

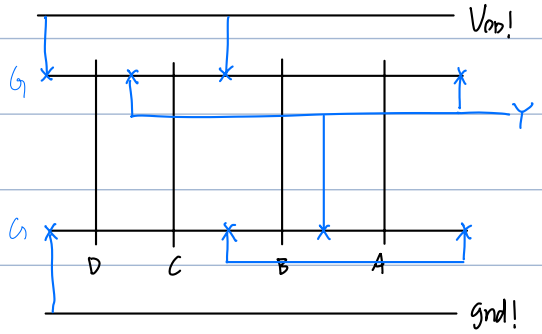


complementary static CMOS

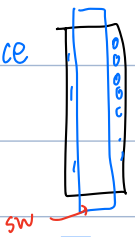
Euler path: runs every M once,

pun/pdn same order input

CCC/CCR (channel connected component region)



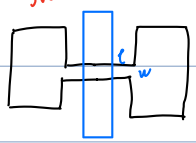
Wide device



issues: poly interconnect resistive → RC delay ↑. Top switches slow

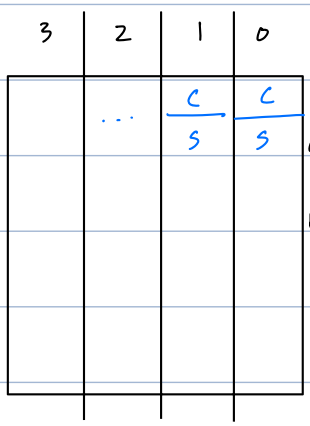
sln. break into parallel (n-fingers) (if > 10 um → finger)

Dogbone



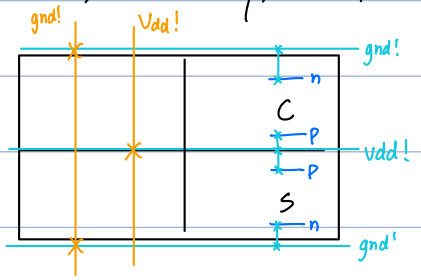
for very small w

Layout (bitstack design)

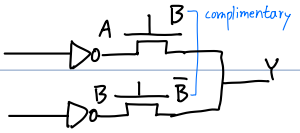


Same block, height varies by complexity

↕ data, ↔ carry, shift, ···



Pass transistor: MOS use as generic switch

E. 

A, B used to simulate real outputs (instead on Vin)

static, since always on path passed

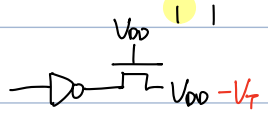
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

→ AND

problem 1. subthreshold leak may affect

2. loses overdrive when up →

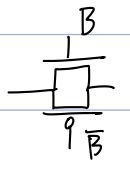
3. only to $V_{DD} - V_T$ → lose NM

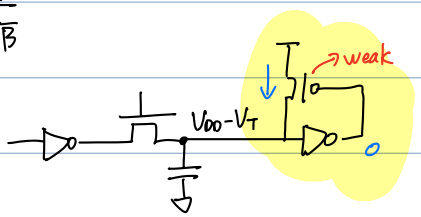


Fix 1. complimentary pass transistor

when pulldown, N all the way ↓; ··· PTT

but complimentary signal generation may not be easy

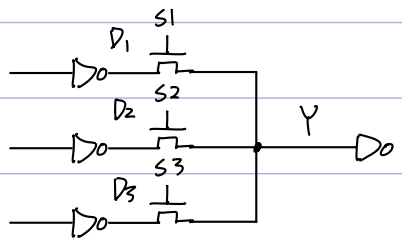




2. Half latch: weak PMOS pulls all to V_{DD} → NM↓

but when pulldown, need to fight half latch pulling V_{DD}

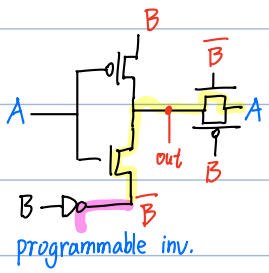
E. MUX static only if one-hot (logically orthogonal)

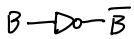


E. 6-transistor XOR

B=0 → PT on, prog. inv. off , Y=A

B=1 → inv. on , $Y = \bar{A}$





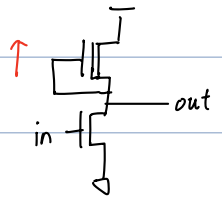
(4+2)

ratio logic: only works if the sizing is right

E. A=1, B=0→1, prev. inv. N pulldown, but if pullup overpowers, :(

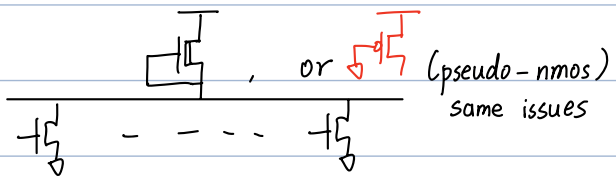
Nonstatic

1. ratio logic (pu, pd simul.)

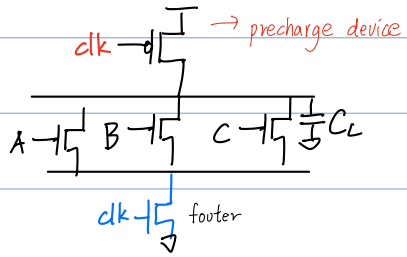


NMOS logic (enhance + deplete)

- depletion acts as static pullup. pulldn enhance need to overpower it
- slow pulldown, wasteful
- Advantage very wide NOR
- Need PMOS stack, but only one dep. N.



2. Dynamic logic (none on → Z)



clk=0: precharge, make all inputs 0 → $Y = V_{DD}$

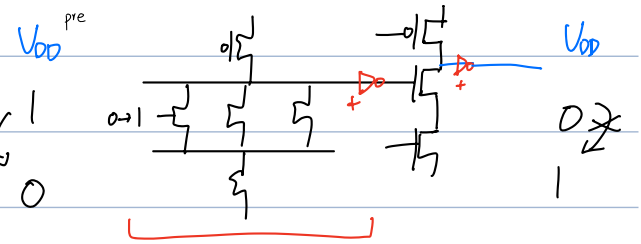
clk=1 evaluate, if all 0 → Y=1; else Y=0

↳ precharged and held, Z

No resistance (fight) @ pulldown (discharge C_L), fast. One-time pulldown (one-short hazard-free)

issue 1. Add a footer Q, but more stack height → delay ↑ (input can ≠ 0 if clk=0)

2. driving capacity E. or gate (dynamic)



if all start at 0, output discharged

then input ⌐, out can't recharge → race cond.

sln. buffer w/ 2 inv., so stage 2 input =0 @precharge

aka domino logic (precharged like domino, monotonic eval)

basic domino gate is not inverting → incomplete :(→ gate need dual rail logic + bubble push

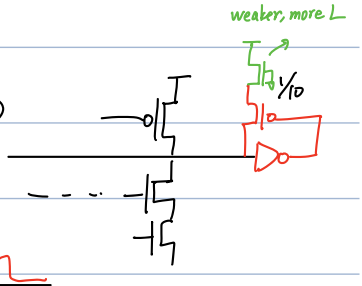
logical effort: in static CMOS, need P/N network.

in dyn., no pullup! → less C_g total. No crowbar I when pulldown; avoid p

Noise dynamic node (out) very noise sensitive. Z mode only stores at C_g → sensitive.

digital noise mainly interference

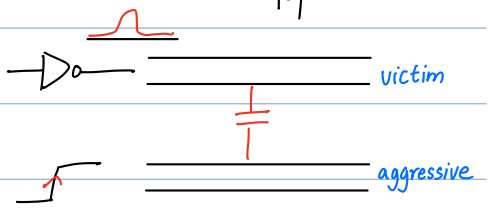
1. Leakage E. many NORs, precharged to 1, all inputs 0 → Z!, but all leak ⇝ 0



Fix w/ half-latch → NM vs perf.

To make weak: W↓ or (L↑, but $C_{g,half}$ ↑) or make a $^{+}$stack

2. Coupling parallel wire → Cap



if aggressive net sw, bilp will fight victim driver.

if ↓ (output) driver is dyn. → 0→1, can't pull back to 0.

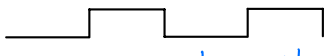
dyn. inputs may also be transiently on ↑

Fix 1. ensure dyn. isn't strongly coupled.

2. half latch (again)

3. Supply noise when pulling current, LRC in V_{DD} - gnd :(

a lot noise @ clk freq. (especially for sync. noise)



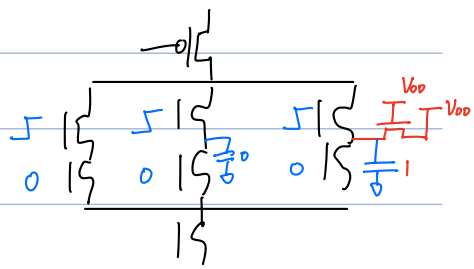
if V_{DD}↓ @ precharge → eval not all the way charged

V_{DD}↑ @ eval → eval < $V_{DD\,now}$ → wrong! (want less 10%)

Also when V_{DD}↓, slow (less V_{DD})

4. Charge sharing noise (solved by half latch)

When inputs ⎍, charge shared w/ internal C_a nodes → eval V↓

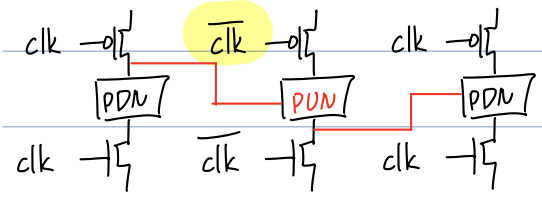


or 1. add babysit devices (bleeder nfet tied to Vdd) →

2. pfet precharge internal nodes

→ more charge to switch → perf ↓

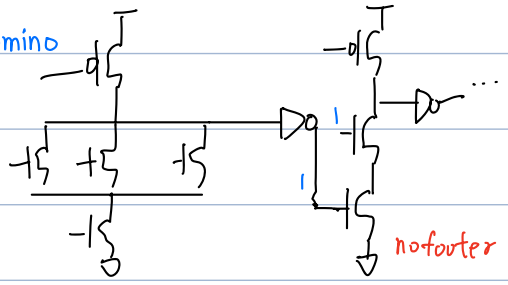
NP domino – regular domino needs D0 to avoid race condition



1 won't turn on pfet, 0 won't turn on n

:(p worse, not too bad in modern tech

Delay reset domino

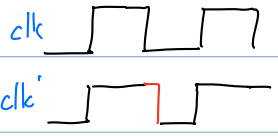


the clks enter precharge at some time.

When 1st stage outputs 1, → short circuit

sln. delay precharge of clk'

duty cycle ↑ → delayed pre.



want to give stage 1 long enough time to pre before stage 2 starts to pre.

However, need + delay for every stage → :(→ restore w/ footed device + clk

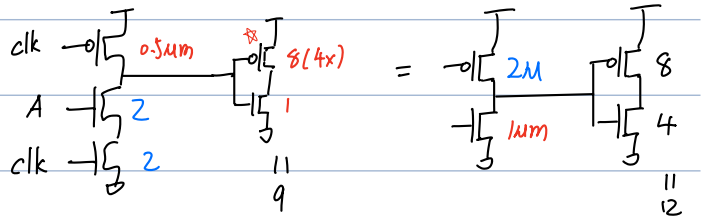
Skewed logic Can size pfet for clk, nfet D0 down → saves C. Just enough to pre.

dyn. ↑ put all on one transition (n) → cut p

Logic effort (at A) $\frac{2}{3}$ compared to push-pull

(at D0): $\frac{9}{12} = \frac{3}{4}$ (generally x4)

(all delays normalized to FO1, which is push pull)



E. drive a 25 um cap, each stage w/ gain of 3.

$W_{pD0} = \frac{25M}{3 \cdot 3} \times 2 = 5.6\,um$ (gain, nfet D0, pfet), $W_{nD0} = 0.7\,um$

$W_{nin\ nclk} = \frac{5.6+0.7}{3 \cdot 3} \times 3 = 2.1\,um$ (3-high stack), $W_{pclk} = 0.7\,um$